

## **DOD 5000.4-M-2 SOFTWARE RESOURCES DATA REPORT (SRDR) MANUAL**

### **CHAPTER 3. INSTRUCTIONS FOR THE DD FORM 2630 SERIES SOFTWARE RESOURCES DATA REPORT (SRDR)**

#### **3.1 Introduction**

The forms in the DD Form 2630 series are used to describe the development or upgrade of a major software element. The DD Form 2630 series is collectively titled the Software Resources Data Report (SRDR). Any submission of a report in the DD Form 2630 series must be accompanied by an explanatory document, known as a SRDR Data Dictionary, which explains data definitions and any details required to correctly interpret the responses. The described software development or upgrade effort can be the subject of a single software contract, a deliverable release within a larger software effort, or a software component of a larger system contract. The subject development or upgrade can be performed commercially or as an internal (“organic”) DoD effort.<sup>1</sup> The DD Form 2630 is designed to record both the expectations and actual results of new software developments or upgrades. It is not designed for reporting on, nor should it be used for, software maintenance or software operation and sustainment efforts. Similarly, the reporting form should not be used for collecting management tracking measures during the course of a project since the sample data items are not designed to record partial progress or interim results.

This document explains the content of the DD Form 2630 series by describing each data item contained in the sample forms shown in Chapter 2. The data items shown on the sample forms are only examples and must be customized to be consistent with data that the development organization normally maintains to manage a project and also to be in accordance with the approved Software Resources Data Collection Plan, developed by the Cost Working-level Integrated Process Team (CWIPT). Thus, the sample forms illustrate but do not mandate the data items needed to satisfy the basic requirement to estimate and report software size, effort, schedule, and (optionally) quality at the beginning and end of a major software development or upgrade.

This chapter constitutes a set of instructions for the sample forms, showing the level of detail that would be needed to explain any customized or added data items. As such, the sections of this chapter can be used as a point of departure for a customized SRDR Data Dictionary. Other than deferring to the CWIPT, these instructions do not specify a process for customizing, completing, or submitting DD Form 2630 forms.

Three instances of the DD Form 2630 are required to record the customer’s and developer’s expectations as well as the actual outcome of a project: a planning report completed by the program office at the time of solicitation (DD Form 2630-1), an initial report completed by the developer at the beginning of development (DD Form 2630-2),

---

<sup>1</sup> For convenience, the term contract is used in this document to mean the authorizing vehicle or agreement that describes the software development or upgrade project whether or not it is in the form of a formal contract.

and a final report completed by the developer at the end of development (DD Form 2630-3). Additional forms are required if the contract consists of multiple releases or constituent elements of software. In this case, separate forms are required prior to development (DD Form 2630-2) and after delivery (DD Form 2630-3) of each release or element.

The government program management office for a reporting project submits an Initial Government Report, DD Form 2630-1, customized as necessary, before contract award (e.g., as part of the Cost Analysis Requirements Document or CARD, due 180 days before contract award). The development organization (e.g., contractor or CDA) submits an Initial Developer Report DD Form 2630-2, customized as agreed upon with the program management office, within 60 days after contract award. The development organization should submit a Final Developer Report, DD Form 2630-3, customized as agreed upon, within 60 days of final delivery describing the as-delivered software product and its development process. In the case of multiple incremental deliveries (builds, releases, versions, elements, etc.), the development organization should submit, within 60 days of the start of any increment, and additional DD Form 2630-2 containing estimates for that increment. The development organization should then submit, within 60 days of delivery of an increment, an additional DD Form 2630-3 describing the as-built product and its development process.

It is assumed that forms will be submitted as computer files (Excel readable) in order to allow convenient customization of the names and numbers of data items. Each DD Form 2630 series form must be submitted with a similarly customized SRDR Data Dictionary. The sign-off area on page two includes space to identify the file name and revision for the associated SRDR Data Dictionary.

Each sample DD Form 2630 series form is divided into two pages. Page one has three sections (Section I, II, and III). Page two has two additional sections (Section IV and V) plus a sign-off area at the end. Space for brief comments, explanations, or context information is provided after each part. More extensive comments should be documented as part of the associated data dictionary.

### **3.2 Instructions for Part 1: Report Context**

Items 1 through 4 of Part 1 should be completed for all three submissions of the DD Form 2630. Additional items (5 through 10) are to be completed after the development organization has been identified (DD Form 2630-1 and DD Form 2630-2, only).

#### **1. System/Element Name (version/release)**

This is the name used to refer to the software product being developed, including any applicable version, release, build, or other identifier. Include the name of the work breakdown structure (WBS) element and its associated WBS number.

#### **2. Report As Of**

This is the date as of which all other answers are meaningful for this submission of the form. If a subsequent report supersedes a previous report, for example to correct an

error, this date would be the retroactive date of the superseded report rather than the current date.

3. Authorizing Vehicle (MOU, contract/amendment, etc.)

This is the contract number (if applicable) and amendment number (if applicable), or reference to a memorandum of understanding or other documentation that authorizes the development of the subject software.

4. Reporting Event

The event that drives this submission of the DD Form 2630 is already shown in the sample customization. Possible choices are, "CARD," "Project/Release Start," or "Contract/Release End" corresponding to the DD Form 2630-1, 2630-2, or 2630-3, respectively. Space is provided to indicate the specific submission number of this form, so as to identify it in the event that a subsequent form is needed to correct or revise an earlier submission.

5. Development Organization

For report submissions after contract award, this is the name of the company or organization that is the responsible developer of the software product being developed. The associated SRDR Data Dictionary should be used to explain the mapping of development organizations, software components and DD Form 2630 forms submitted. As with any other customization of this form, agreement on the level of aggregation must be reached between the developer and program office.

6. Certified CMM Level (or equivalent)

This is the Software Engineering Institute (SEI) Capability Maturity Model (CMM) number of the level (1 through 5) at which the primary development organization has been formally certified. If no formal certification has been conducted, leave the item blank. If a single submission is used to represent the work of multiple organizations, enter the level of the organization that will be expending the most amount of effort on the development project (not necessarily the prime contractor) and note this in the associated SRDR Data Dictionary. If the government has accepted an alternate assessment mechanism, such as the SDCE (Air Force) or ISO-15504, enter a pointer to the results here and explain the meaning of the assessment in the SRDR Data Dictionary. It is possible for this assessment to change between an initial developer and a final developer submission.

7. Certification Date

If the answer to item 7 is non-blank, this is the date when the formal assessment associated with the indicated level was conducted.

8. Lead Evaluator

If the answer to item 7 is non-blank, this is the name of the person that lead the formal SEI CMM assessment and determined the maturity level indicated.

9. Affiliation

This is the affiliation of the Lead Certifying Analyst in the previous item.

#### 10. Precedents

Up to five analogous systems that have been developed by the same software organization or development team are listed here.

### 3.3 Instructions for Part 2: Product Description

Most of the items in Part 2 are included on all three forms of the DD Form 2630 series. Only the development process and developer experience are omitted from DD Form 2630-1 (initial government report). The numbers for these items are skipped in the sequence on that form so that other items have numbers that correspond to their counterparts.

#### 1. Primary Application Type

Using one or more domain names from the list in section 3.7 of this chapter, when possible, describe the primary application type being developed. The primary type may be the only application type listed, but any number of application types may be listed. (Space for four is provided on the form but submissions may include any number.) If none of the examples shown in the list of application types are appropriate, enter a phrase to describe the application type and define it in the associated SRDR Data Dictionary. When there are internal development efforts within a program that are large and independent, respondents may choose to report each using a separate DD Form 2630 instead of as various application types within a single report.

#### 2. Percent of Product

This is the approximate percentage of the product size that is of the indicated primary application type, up to 100%.

#### 3. Development Process

For the initial developer DD Form 2630-2 and final developer DD Form 2630-3 submissions, this is the name of the development process planned or followed for the primary application of the system. Use common industry terms, such as waterfall, spiral, or RAD, rather than proprietary names that are internal to the development organization. Do not indicate a software architecture method (such as object-oriented development) or a development tool (such as Rational Rose), as these do not specify a process.

#### 4. Upgrade or New

This indicates whether the primary development is new software or an upgrade. A software system is considered new either if no existing system currently performs its function or if the development completely replaces an existing system. A software system that replaces part of an existing system (such as the replacement of a database) should be considered an upgrade. An existing software system that is being ported to a new platform or being reengineered to execute as a web or distributed application (for example) would be considered an upgrade unless it is also being completely redeveloped from scratch (new requirements, architecture, design, process, code, etc.).

### 5. Secondary Application Type

If the development contains a major secondary application type, indicate it here.

### 6 - 8. Secondary Application Type Details

This indicates the system percentage of the secondary application type, its development process and whether it is new or an upgrade.

### 9 - 12. Third Application Type and Details

This indicates the third application type, its percentage of the system, its development process and whether it is new or an upgrade.

### 13-16. Fourth Application Type Details

This indicates the fourth application type, its percentage of the system, its development process and whether it is new or an upgrade. If a project includes more than four application types, extend the form or submit additional sheets as required.

### 17. Primary Language

This is the computer language in which most of the development is expected to be (or was) conducted. This can be a compiled language, such as FORTRAN, Ada, or C, or it can be an interpreted language, such as Forté. Use the amount of effort spent in development to determine the primary language rather than the amount of function delivered. Explain any interpretation of this item in the associated SRDR Data Dictionary.

### 18. Percent of Product Size

This shows the approximate amount of the final development effort that is expected to be (or was) involved with producing code in the Primary Language. This may differ somewhat from the percent of the final physical product that will be written in this language since a large portion of the delivered product might use generated code or COTS products that are not directly developed.

### 19. Secondary Language

This shows the secondary language used in the development (if any), using the same definitions given under the Primary Language.

### 20. Percent of Product Size

This shows the approximate amount of the final development effort that will be (or was) involved with producing code in the Secondary Language. This may differ somewhat from the percent of the final physical product that will be written in this language since a large portion of the delivered product might use generated code or COTS products that are not directly developed.

### 21. List COTS/GOTS Applications

This shows the names of the applications or products that will (or do) participate in the final delivered product, whether they are commercial off-the-shelf (COTS) or

Government off-the-shelf (GOTS) products. If a proprietary application or product that is not generally commercially available will be (or was) included, identify it here and include any necessary explanation in the associated SRDR Data Dictionary.

22. Peak staff (team size in FTE) expected to work on and charge to this project

This is the expected or actual peak team size, measured in full-time equivalent staff. Only include direct labor in this calculation unless otherwise explained in the associated SRDR Data Dictionary.

23. Percent of Personnel by experience level in domain

For the initial and final reports, this is the percent of project personnel that is expected to be (or was) highly experienced in the domain (three or more years of experience), nominally experienced in the project domain (one to three years of experience), and entry level (zero to one year of experience). The percentages reported at each level should take into consideration the duration each person works on the project (so that, for example, a single highly experienced person who works on the project for two years constitutes the same percentage of the total as two entry level people who each contribute a year of effort). The experience level of a person is rated as he or she begins work on the project or the increment being reported, so that experience gained between the initial and final reports of a project or increment is not counted towards the rating.

### 3.4 Instructions for Part 3: Product Size Reporting

Part 3 asks for quantitative information about the size of the software development. If this is an initial, DD Form 2630-2, provide estimates-at-complete for the relevant release or delivery. If this is a final, DD Form 2630-3 then provide actual values for the delivery or release covered by this report.

1. Number of Requirements, not including External Interface Requirements

This is the number of requirements satisfied or to be satisfied by the developed software product. In the initial reports (DD Form 2630-1 and 2630-2), provide estimates of the total number of requirements to be implemented by the software being developed. In the final DD Form 2630-3, provide the actual number of requirements implemented by the developed software using the same counting method as was used in the estimating reports. Do not count requirements concerning external interfaces not under project control. Explain any details about the requirements counting methods in the SRDR Data Dictionary.

2. Number of External Interface Requirements

This is the number of *external* interface requirements not under project control that the developed system will satisfy. External interfaces include interfaces to computer systems, databases, files, or hardware devices with which the developed system must interact but which are defined externally to the subject system. In the initial reports (DD Form 2630-1 and 2630-2), provide estimates of the total number of interface requirements to be handled by the software to be developed. If the developed system interfaces with an external system in multiple ways (such as for reading data and also for

writing data) then each unique requirement for interaction should be counted as an interface requirement. In the final DD Form 2630-3, provide the actual number of interface requirements handled by the developed software using the same counting method as was used in the initial reports. Explain any details about the external interface requirements counting methods in the SRDR Data Dictionary.

### 3. Amount of Requirements Volatility encountered during development

As part of the final DD Form 2630-3 report, indicate the amount of requirements volatility using a qualitative scale (very low, low, nominal, high, very high) relative to similar systems of the same type. This should be a relative measure rather than an absolute one in order to understand how initial expectations were or were not met during the course of the software development.

### Code Size Measures

This unnumbered block is used to define the code size measure used in items 4 through 6. A measure other than those listed may be indicated if none of those shown are applicable. The preferred size measures are total physical source lines of code or carriage returns (to be indicated below by “S”), noncommented and nonblank source lines of code (to be indicated by “Snc”), or number of logical source statements (to be indicated by “LS”). If another size measure is being used, provide an abbreviation for it and briefly explain it. For example, unadjusted function points, adjusted function points, object points, feature points, classes, algorithms, or other functional measures could be indicated. Use the SRDR Data Dictionary for longer explanations, if required.

The size measure chosen should allow independent verification of the project size by examining the software products produced by the development. For this reason, unless a post-hoc analysis of functional size will be conducted to compare with estimated function points or other functional size estimates, one of the source code counting methods is preferred as a size measure, where “code” can refer to any hand-edited product such as lines of a computer language or lines in tables used to configure a reusable product. Many models normalize to SLOC, which is a convenient common denominator for describing product size, even if the initial planning is done using another measure, such as function points, objects, classes, screens, algorithms, etc. However, developed code size may be expressed in other terms if SLOC is a meaningless measure of the output for the majority of the programmer effort (such as when developing a web page using an iconographic tool interface). As with other customizations, the selected size measure should be in accordance with the approved Software Resources Data Collection Plan, developed by the CWIPT.

The next three items are intended to capture the size of the system under development by partitioning (exhaustive with no overlaps) the code into three categories. (Any customization of this form should maintain a partitioning categorization to avoid double counting or omissions in the delivered code size measurement.) The configuration control system is assumed to be the repository for completed code. (Unless otherwise explained in the associated SRDR Data Dictionary, code that is developed but not maintained under a configuration control system is not to be considered part of the developed system.) Only the most recent version of each code unit should be counted.

For each of the next three items, indicate the size measure abbreviation in the blank provided.

#### 4. New Code

Most software projects utilize a combination of new, reused, and generated code to accomplish the required function. Any code that was developed specifically for this project, or was reused or generated by tools but then extensively modified (more than 25% of the lines changed or added), is considered new code. Code generator inputs prepared by hand, such as tables or scripts, are also counted as new code.

#### 5. Modified Code

Source code that was generated by tools or obtained from outside the project (even if within the same organization) and was then reused with minor modifications (less than 25% modified) by this project is reported under this item. If modifications were substantial (more than a notional 25%), the code is counted as new (item 4). This assessment should be done at the code unit level and not across the whole project.<sup>2</sup>

#### 6. Reused Code

Source code that was obtained from outside the project (even if within the same organization) or that was generated by tools and not modified at all is reported under item 6.

### 3.5 Instructions for Part 4: Resource and Schedule Reporting

Project development is typically broken down into phases or activities. This form can be customized to include the names of the phases or activities that are appropriate for the subject development.

#### 1 - 6. Software Development Activities

Items 1 through 6 under Part 4 are taken from the activity definitions used in ISO12207 and are intended to be generic to any software development (though they may not be strictly associated with development phases by the same names). These activities may be performed simultaneously, sequentially, or both. The two initial reports (the DD Form 2630-1 and the DD Form 2630-2) include estimates of the schedule and total effort applied to each activity. The final report contains actual schedules and total efforts for each activity. Many of the activities will overlap, even in a waterfall style of development. In an iterative or spiral development, activities may start and stop. To the extent that is sensible for the approach used (or expected), the dates are the earliest and

---

<sup>2</sup> As a simplistic example, if a 100,000-line project consists of 100 units of 1,000 lines each, and 30 of those units each have 100 modified lines (each unit being 10% modified), then that entire collection of 30,000 lines should be considered modified code. However, if another 20 units each have 300 modified lines (each unit being 30% modified), then that entire collection of 20,000 lines should be considered new code.



latest that each activity occurred (or is estimated to occur). Month numbers, starting with month 1 at the time of Contract Award, are shown in the first two columns.<sup>3</sup>

#### 7. Other Direct Software Engineering Development Effort

Item 7 is for any direct project hours that are not accounted for in the previous six items. (Schedule is not applicable to this item.) In the text space provided, summarize the kinds of activities included, such as project management, IV&V, configuration management, quality control, problem resolution, library management, process improvement, measurement, training, documentation, data conversion, or supporting a customer-run acceptance test. Also include software delivery, installation, deployment and/or implementation, to the extent these activities are included in the development contract. If any *allocated direct* charges are applied to a project, they should be included in this item.

The contribution of any *indirect* hours is described in the comment block or in the SRDR Data Dictionary (e.g., training, process improvement, methodology research) but not included in these totals.

### 3.6 Instructions for Part 5: Product Quality Reporting (optional)

Desired quality is requested on the program office CARD (DD Form 2630-1) report at part 5, item 1a or 1b. Actual quality of the delivered system is requested on developer final reports (DD Form 2630-3) at part 5, item 2a or 2b. No reporting of estimated quality is needed for the developer's initial reports (DD Form 2630-2). The sample DD Form 2630 suggests quantifying quality operationally (through failure rate and defect discovery rate). However, other methods may be used if appropriately explained in the associated SRDR Data Dictionary. Quality reporting may be deemed inappropriate by the CWIPT. If so, a project may tailor Part 5 out of its DD Form 2630 series reports.

#### 1a. Required Mean Time to Defect (MTTD) at Delivery

The required MTTD at time of delivery is one method by which a customer can specify nominal product quality. The definition of this measure must include whether minor or only major (mission compromising) defects are counted, and whether recurring known defects or only new ones are counted. Also, the operational time basis must be clarified, such as by indicating whether a system is only operational eight hours a day or continuously, or whether a system operates in a single instance or in multiple instances at different locations simultaneously. Use the associated SRDR Data Dictionary to clarify the counting method.

#### 1b. Analogy with Similar Systems

An alternative method to specify nominal quality is to compare the required reliability of this system with typical reliability for systems of this type. For example, if

---

<sup>3</sup> For builds or releases that do not begin at the start of a project, such as a build subsequent to an initial build, the starting month number can be greater than 1 for schedule estimation or reporting purposes.

the system is an operational flight program (as noted in Part 2, item 1), higher than nominal reliability might be expected for the OFP of a fly-by-wire aircraft.<sup>4</sup> On the other hand, if the OFP were to control a pilotless vehicle, such as a surveillance or drone aircraft, the required reliability might be lower than average for OFP systems. A customization of this item could allow the response to be in terms relative to other similar systems, for example a scale such as “much higher,” “somewhat higher,” “nominal,” “lower,” or “much lower” would be appropriate. As with any customization, the explanation of the data must be included in the SRDR Data Dictionary.

#### 2a. Measured or Computed Mean Time to Serious or Critical Defect (MTTD)

At Contract End, an actual measure of software quality can be reported. The DD Form 2630-3 includes items 2a and 2b as two examples of how delivered product quality may be reported. Item 2a is an example of a quantitative measure of quality using the observed or computed interval between serious or critical defect discoveries. (An example of five defect categories can be found in the superseded MIL-STD-498. Developers may customize these definitions to conform to their existing definitions.) Developers should use existing procedures for distinguishing defects from routine development changes, such as problems found after an inspection, after a configuration control baseline, or after advancement to the next state of a development process.

#### 2b. Analogy with Similar Systems

Item 2b is an example of a qualitative measure of product quality using analogy to other similar systems. Use the SRDR Data Dictionary to document details of this or any other quality measure used.

### Filename and Revision Date of Applicable Software Resources Data Report Data Dictionary

The definitions of any customized item or any other clarifying definitions of metrics reported on a submitted DD Form 2630 should be contained within a SRDR Data Dictionary. Submitters are encouraged to submit both the DD Form 2630 and the SRDR Data Dictionary as electronic files. The name and date of the file containing the data definitions should appear here.

### Point of Contact and Sign Off

The form concludes with a sign-off line for the name, phone, and e-mail of the contact person to handle any inquiries about the data submitted, plus the date of completion (which would usually be later than the as-of date in part 1).

## 3.7 Application Types

Use the following domain names (mission and function areas) in Part 2 of the DD Form 2630 to specify the application type(s) for the software system under development. If none of these domain areas are applicable, enter a phrase that describes the application type and define it in the associated SRDR Data Dictionary. The following list contains

---

<sup>4</sup> See also section 3.7 Application Types, at the end of these instructions.

descriptions of overlapping areas; it is not an attempt to partition the possible domain space. It is preferable to identify sufficiently general domains in Part 2 of the DD Form 2630 to avoid the need to use a large number of different application types to describe the system.

### 3.7.1 Warfare Mission Areas

- Antiair Warfare
- Antisubmarine Warfare
- Naval Antisurface Ship Warfare
- Amphibious Warfare
- Chemical Warfare
- Biological and Radiological Defense
- Land Warfare
- Special Warfare
- Strategic Warfare
- Tactical Air Warfare
- Electronic Warfare
- Strategic Defense Initiative

### 3.7.2 Mobility Mission Areas

- Air Mobility
- Land Mobility
- Sea-Surface Mobility
- Undersea Mobility
- Space Mobility

### 3.7.3 Communications, Command & Control/Intelligence Mission Areas

- Communications, command & Control
- Intelligence, including Reconnaissance

### 3.7.4 Mine and Obstacle Mission Areas

- Land Mine/Obstacle/Countermeasures
- Sea Mine/Countermine

### 3.7.5 Mission and System Support Mission Areas

- Logistics
- Manpower, Personnel and Training
- Mission/System Support

### 3.7.6 Weapon Systems Functions

- Target Acquisition/Search/Detect
- Threat Evaluation
- Target Tracking
- Weapon Assignment
- Fire Control Acquisition and Designation
- Launch
- Propulsion
- Control
- Flight Controls
- Conventional munitions/Weapons
- Directed Energy Weapons
- Hard Target Kill/Anti-Armor
- Fuzing
- Chemical Warfare (Offense)

### 3.7.7 Defensive Systems Functions

- Hit Avoidance
- Signature Control/Suppression Reduction
- Armor, Infantry and Crew Protection
- EMP Hardening/Survivability from Nuclear Weapons
- Damage Control
- Chemical/Biological Defense
- Deterrence

### 3.7.8 Mine Functions

- Mine Mooring
- Mine Neutralization/Destruction

### 3.7.9 C3I Functions

- Information Management
- Communication
- Guidance/Navigation/Position Location
- Avionics/Vetronics/Display Systems

### 3.7.10 Electronic Warfare Functions

- Electronic Countermeasures
- Jamming
- Deception
- Cryptography
- Electronic Counter Countermeasures
- Low Probability
- Electromagnetic Signal Measurement/Intelligence

## Jam Resistance

### 3.7.11 Assessment/Analysis Functions

- Simulation

- Weapons and Munitions Effects/Target Kill Assessment

- Vulnerability Analysis

### 3.7.12 RDT&E Functions

- Energetic Materials

- Manufacturing Technology

- Electronics

- Other than Electronics

- Materials Development

- Metals, Ceramics, Organics and Composites

- Electronics

- Test Equipment/Technology

- Structural

- Electronics

- Reliability

- Maintainability

- Structures, including Design and Manufacture

- Missile

- Aircraft

- Hull

- Body/Chassis

### 3.7.13 Miscellaneous Functions

- Multi-Function Applications

- Robotics

- Human Factors/human Engineering

- Artificial Intelligence/Adaptive Systems

- Basic Scientific Research/University Interactions

### 3.7.14 Supply/Support/Construction Functions

- Material Distribution and Payload Handling/Supply Systems

- Training

- Field Services (Water, Food, Tents, etc.)

- Bridging/Obstacles

- Support and Auxiliary Equipment

- Habitability

- Environmental Effects

- Facility Construction

### 3.7.15 Management/Personnel Functions

- RDT&E Management
- Acquisition Management
- Financial Management
- Medical/Casualty Care
- Performance Appraisal

### 3.7.16 Other Embedded Functional Areas

- Avionics
- Audio signal processing and enhancement
- Command and Control
- Command, Control and Information
- Command, Control, Communications and Information
- Command, Control, Communications, Computers and Information
- Digital Signal Processing
- Guidance and control
- Image processing and enhancement
- Operational Flight Program
- Simulation
- Telemetry
- Target seeking
- Embedded trainer software
- Embedded Weapon

### 3.7.17 Other Software System Functions

- Decision Support
- Financial, Accounting, Bookkeeping, Payroll, etc.
- Information System
- Management Information System
- Personnel, Human Resources, etc.
- Operating System
- Online training or education software